Department of Computer Science and Engineering, NITK, Surathkal

Course Plan and Evaluation Scheme

(B. Tech 2nd Semester, 2020-21)

1. Course code: CS111

2. Course Title: Computer Programming Lab

3. L-T-P: (0-0-2)

4. Credits: 1

5. Semester: 2nd Semester B. Tech.

6. Academic Year: 2020-21

7. Teaching Department: Computer Science & Engineering

8. Objective of the Course:

- Learning to code and debug C programs in a UNIX environment.
- Implement programming strategies to understand various C language concepts.

9. Skill development expected from the course:

- Develop efficient and modular C code.
- Develop programming skills, logical thinking and reasoning for problem solutions.

10. Course Coverage:

Module No.	Topic	No. of Labs	Remarks
M-1*	UNIX shell commands, execution, and debugging of programs	1	-
M-2	Basic input-output, operators, and expressions	2	Submission of the report (consisting of exercises' source code and snapshots of the results) in pdf format is compulsory.
M-3	Decision making, branching, and looping	2	
M-4	Arrays & Strings	2	
M-5	Functions and Recursions	2	
M-6	Structures and Union	2	
M-7	Pointers	2	
M-8	Files and file operations	1	

Note: * - No submission of the report

11. Course Contents

M-1. UNIX shell commands, execution, and debugging of programs

Introduction to basic commands of writing, compiling and executing C programs

M-2. Basic Input-Output functions

1. To find Simple and Compound Interest
2. To read the radius of a circle and find its Area and Perimeter.
3. To read the temperature in Fahrenheit and convert it to degree centigrade.
4. Program to accept student roll no, marks in 3 subjects and calculate total, average of marks and print them with appropriate messages.
5. An Employee's Basic Pay (BP) is to be read through a keyboard. DA is 40% of BP, HRA is 20% of BP, calculate the Gross Pay (GP) GP is computed as BP+DA+HRA.
6. Program to find the distance between two points (x1, y1) and (x2, y2) in a Cartesian plane.
7. Program to swap two numbers using a temporary variable. Also print the original and exchanged values.

M-2. Operators and Expressions

(Program to illustrate the use of Arithmetic, Relational, Logical, Assignment, Conditional, Increment or Decrement, Bitwise, Special operators, their associativity and precedence.)

1. The distance between two cities (in km.) is input through the keyboard. Write a program to convert and print this distance in meters, feet, inches and centimeters.
2. The length & breadth of a rectangle and radius of a circle are input through the keyboard. Write a program to calculate the area & perimeter of the rectangle, and the area & circumference of the circle.
3. If a five-digit number is input through the keyboard, write a program to calculate the sum of its digits.
4. If a five-digit number is input through the keyboard, write a program to reverse the number.
5. If the total selling price of 15 items and the total profit earned on them is input through the keyboard, write a program to find the cost price of one item.
6. Write a program to compute the values of square-roots and squares of the numbers 0 to 100 in steps 10.

M-3. Decision making- and Branching constructs

1. To determine whether a character entered is in lowercase, uppercase, digit or a special character.
2. Find the roots of quadratic equation
3. Given 3 sides, write a program to check whether the triangle can be formed for the following conditions.

No triangle	if $a >= b+c$
Right Angled triangle	if $a^2=b^2+c^2$
Obtuse triangle	if $a^2>b^2+c^2$
Acute triangle	if $a^2<b^2+c^2$
Equilateral triangle	if all sides of the triangle are same
Isosceles triangle	if two sides of the triangle are same
Scalene triangle	otherwise

Implement using switch-case, if and else-if ladder:

4. Write a menu driven program to demonstrate the simple arithmetic calculator
5. Program to display the grade obtained by a student based on the marks. The relation between the grades and marks is –

Marks	Grades
Below 40	Fail
40-59	E
50-59	D
60-69	C
70-79	B
80-89	A
90-100 A+	

M-3. Decision making and looping constructs

1. Program to reverse the digits of a number and to find the sum of the digits.
2. Program to find factors of a given number.
3. Program to find the prime and non prime numbers between a given range.
4. Program to print numeric pyramid (Defined by the instructor).
5. Program to find GCD and LCM of two given numbers.
6. Program to evaluate $1+x^2/2! + x^3/3! +\ldots\ldots.+x^n/n!$

M-4. Arrays

1. To read an array of N elements and reverse it.
2. Program to count the number of palindromes in a given list of n numbers

3. Find the pair of numbers in an unsorted array such that their sum is the largest
4. Program to insert an element in a specified position in a given array.
5. To search for a key in an array using i)Linear search ii)Binary search
6. To sort array elements in ascending order using i) Bubble sort ii) Selection sort.
7. To add and multiply two matrices.
8. To interchange principal and secondary diagonal elements in a matrix.

M-4. Strings

1. To read a string and find its length without using a library function.
2. To reverse a string and check whether the string is palindrome or not.
3. To delete extra blank spaces in a string
4. Program to replace lowercase characters by uppercase & vice-versa
5. Program to delete all occurrences of a substring in a given line of text.
6. C Program to find all consecutive occurrences of any vowel in a string

M-5. Functions

1. Write a function to generate n Fibonacci numbers.
2. Program to implement the following string functions (use switch-case to make the program menu driven)
    - i. Find length of a string.
    - ii. Copy a string to the other.
    - iii. Compare two strings.
3. Program to evaluate the equation $y=x^1+ x^2+ x^3 +..........x^n$
4. To add numbers using function.
5. Program to calculate the value of ^np_r.
6. Program to find whether an entered year is a leap year or not.

M-5. Recursion

1. To find factorial of a number
2. To generate Fibonacci series up to a given number.
3. To search for a number using binary search method
4. To find the sum of series 1+1/3!+1/5! + …… +1/N!
5. Program to reverse a string.

6. Program to find the biggest number in an array.

M-6. Structures and Union

1. Program to input and display book information (Title of the book, Author , ISBN, Price)
2. Program to accept 5 people's name, address and telephone number and to search for the information of a particular person.
    i. Based on name
    ii. Based on telephone number
3. Accept a name, register number, marks in 6 subjects of N students. Find the total and average of each student. Sort this array of structures
    i. Based on name.
    ii. Based on register number.
    iii. Accept a register number and search for this student using binary search
    iv. Find the average marks for each student and sort them in ascending order of average marks.
    v. Display the details of a student given his register number
4. Store the item number, item name, unit price and quantity in stock of N items in a supermarket. Display the following list of items present in the stock.
    i. List of items with unit price greater than Rs 129.
    ii. List of items with quantity in stock less than 5.
5. Write a menu driven program for the following
    i. Add two distances (in inch-feet) using structures
    ii. Add two complex numbers by passing structure to a function
    iii. Calculate the difference between two time periods using structures.
6. Write a C program to compare structure and union variables

M-7. Pointers

1. Program to count frequency of every character present in a line of text.
2. Program to swap two numbers.
3. Program to find area and circumference of a circle.
4. Write functions for the following string operations.

a) Concatenation. b) Comparison. c) Length d) Copy e) Reverse.

5. Write a program to display the greatest of N numbers –use malloc() function.
6. Write a program to arrange N names in alphabetical order using dynamic memory allocation.

M-8. Files and file operations

1. Program to read data from the keyboard; write it in a file called DATA.txt. Again read the same data from the file and display it on the screen.
2. A file named DATA.txt contains a series of integer numbers. Write a program to read these numbers and then write all odd numbers to a file to be called ODD and all even numbers to a file to be called EVEN.
3. Copy the content of one file into another.
4. Program to append the content of a file at the end of another without using append mode.
5. Program to find the size of file using file handling function.
6. Program to find the number of lines in a text file.

12. Evaluation Plan:

Sl. No.	Items	Weightage (%)	Remarks
1	End-Sem	35	Common to all 14 sections
2	Mid-Sem	20	Common to all 14 sections
3	Regular Lab Performance	15	Left to the individual Instructor
4	Class Test	30	Common to all 14 sections

Note:

- Grade cutoff is common for S1-S14 sections.
- A common end-sem, mid-sem and class test-1 & -2 will be conducted **through IRIS** for S1-S14 section as per the following tentative schedule.

Sl. No.	Items	Date	Time	Syllabus
1	Class Test-1	04.06.2021 (Friday)	05:00 pm - 05:30 pm	M-2 and M-3
2	Mid-Sem	25.06.2021 (Friday)	05:00 pm - 06:00 pm	M-2, M-3, and M-4
3	Class Test-2	23.07.2021 (Friday)	05:00 pm - 05:30 pm	M-5 and M-6
4	End-Sem	12.08.2021 (Thursday)	05:00 pm - 06:30 pm	M-2 to M-8

- Missing labs due to holidays should be compensated by conducting extra labs during the same week and should be announced well in advance to all students.

Lab Instructions (Procedure):

1. Every lab module (M-1 to M-8) starts with the explanation and demonstration of sample programs by the lab instructor along with doubt clearing during the online lab session scheduled as per the time-table.
2. The second subsequent online lab will be used to evaluate the exercises completed by a student of that particular module. At least 4 teams for every section consisting of 3 members (AL, RS, M. Tech(R), M. Tech) can be formed to evaluate the students' work. **Supporting teams** details can be found here.

   Example:

   Module M-1 (no report submission) explanation with the demo will be done by the respective lab section instructor during 3rd May - 7th May 2021 (online lab of the respective sections).

   For module M-2 onwards, one lab will be used for explanation and demo by the respective section lab instructor, followed by the subsequent lab for the evaluation. (Example: Section-1, module M-2 explanation and demo will be on 10th May 2021 and evaluation will be on 17th May 2021. Further, on or before 16th May 2021 section-1 students' should upload the report **through the IRIS**.)
3. Regular evaluation lab's **MS Teams link** should be updated here by the RS which will be monitored by the lab instructor.
4. Regular lab **evaluations (marks)** should be updated regularly here by the lab

instructor. The **guidelines for regular lab evaluations** can be found here (not for students).

5. Every student will be given a week's time to complete at least **6 exercises: 3 from each submodule** (extra can be used for practice) listed in a particular module (refer to sl. no. 11) and upload a pdf file consisting of source code and snapshot of the results with the file name <reg. no.>-<Module-No.>.pdf. (Eg. 21CS01-M1.pdf).

13. Useful Links:

1. Lab time-table: Click Here
2. In-Sem provisional evaluation: Click Here
3. Supporting teams : Click Here
4. Supporting teams details: Click Here
5. MS Teams link for evaluation: Click Here
6. Regular lab evaluation guidelines: Click Here (Not for students)
7. Weekly schedule: Click Here
8. Feedback form for students: Click Here

Course Instructors **Head of the Department**